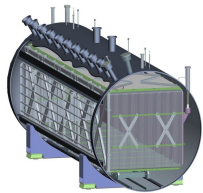


Deconvolution with Induced Charge: Outline of CalROI Scheme

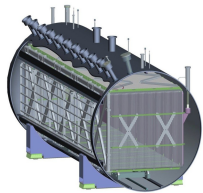
Michael Mooney

*MicroBooNE BNL Meeting
February 25th, 2015*

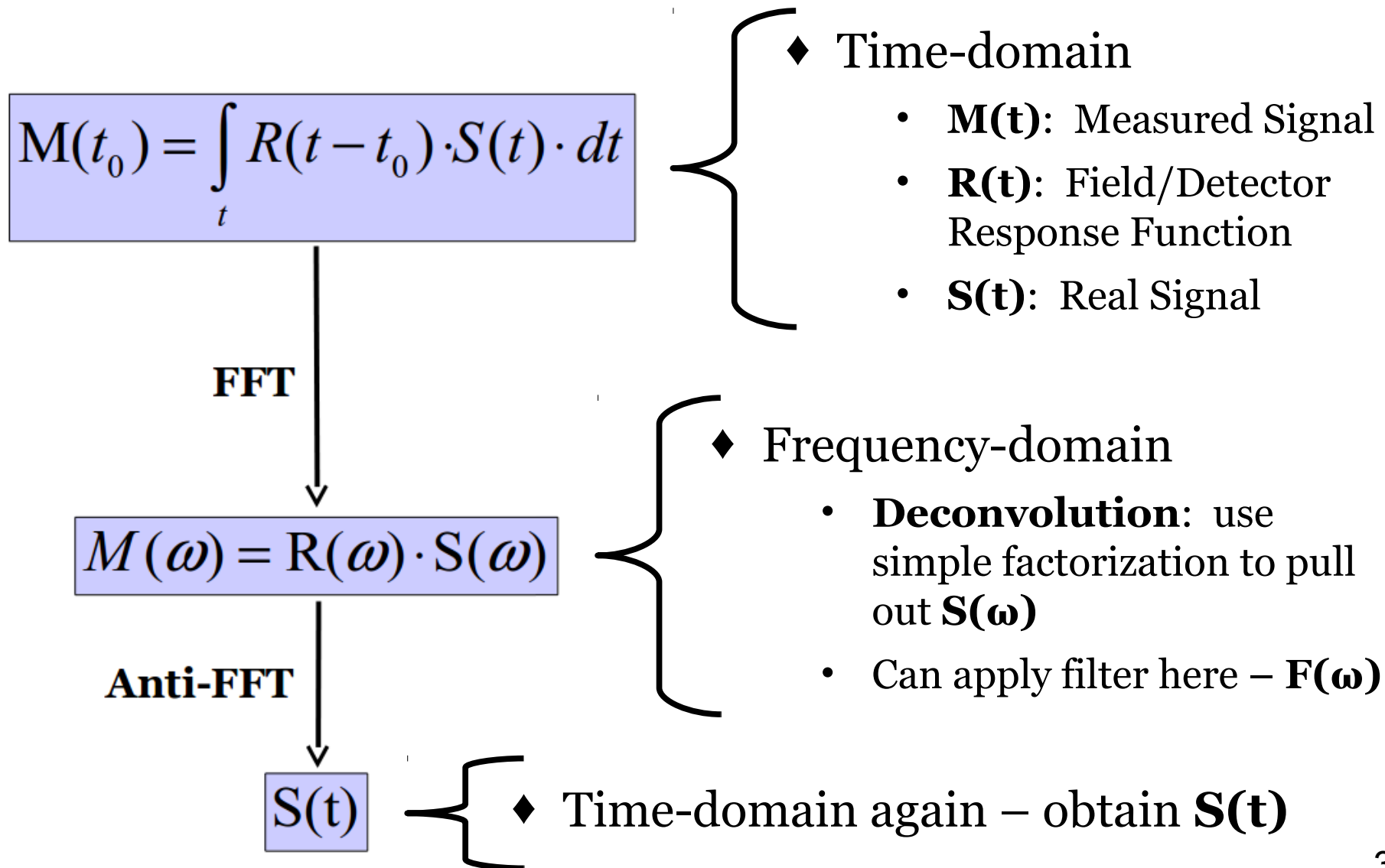


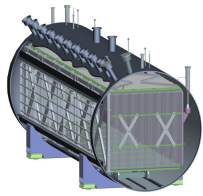
Introduction

- ◆ So far have implemented simple algorithm for including effects of induced charge from adjacent wires within deconvolution
 - Scheme is appropriate for implementation into **CalWire**: use all time bins (9600) and wires (3500/2500/2500)
 - **Twice as slow** as without induced charge
- ◆ Now let's look at an implementation suitable for using with **CalROI** – should be much faster
 - Requires primitive tracking – fairly complex problem!
 - Factorize into several smaller problems, attack one at a time:
 - Merging of ROI's into “**ROI clusters**”
 - Splitting of ROI clusters into smaller units consistent with track/shower candidates – “**ROI subclusters**”
 - For each ROI subcluster find appropriate FFT window across all associated wires – “**ROI boxes**”



Deconvolution Review





New Deconvolution Scheme

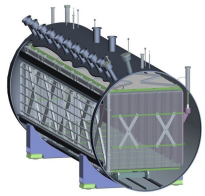
- ◆ Including effects of induced charge from adjacent wires, things become more complicated:

$$M_i(t_0) = \int_t (R_0(t-t_0) \cdot S_i(t) + R_1(t-t_0) \cdot S_{i+1}(t) + \dots) \cdot dt$$

$$M_i(\omega) = R_0(\omega) \cdot S_i(\omega) + R_1(\omega) \cdot S_{i+1}(\omega) + \dots$$

- ◆ Can represent in matrix form:

$$\begin{pmatrix} M_1(\omega) \\ M_2(\omega) \\ \dots \\ M_{n-1}(\omega) \\ M_n(\omega) \end{pmatrix} = \begin{pmatrix} R_0(\omega) & R_1(\omega) & \dots & R_{n-1}(\omega) & R_n(\omega) \\ R_1(\omega) & R_0(\omega) & \dots & R_{n-2}(\omega) & R_{n-1}(\omega) \\ \dots & \dots & \dots & \dots & \dots \\ R_{n-1}(\omega) & R_{n-2}(\omega) & \dots & R_0(\omega) & R_1(\omega) \\ R_n(\omega) & R_{n-1}(\omega) & \dots & R_1(\omega) & R_0(\omega) \end{pmatrix} \cdot \begin{pmatrix} S_1(\omega) \\ S_2(\omega) \\ \dots \\ S_{n-1}(\omega) \\ S_n(\omega) \end{pmatrix}$$

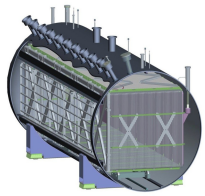


Inverting Response Matrix

- ◆ New deconvolution scheme implementation requires inversion of large response matrix – **too slow?**
- ◆ Can ignore wires that are far away to simplify matrix:

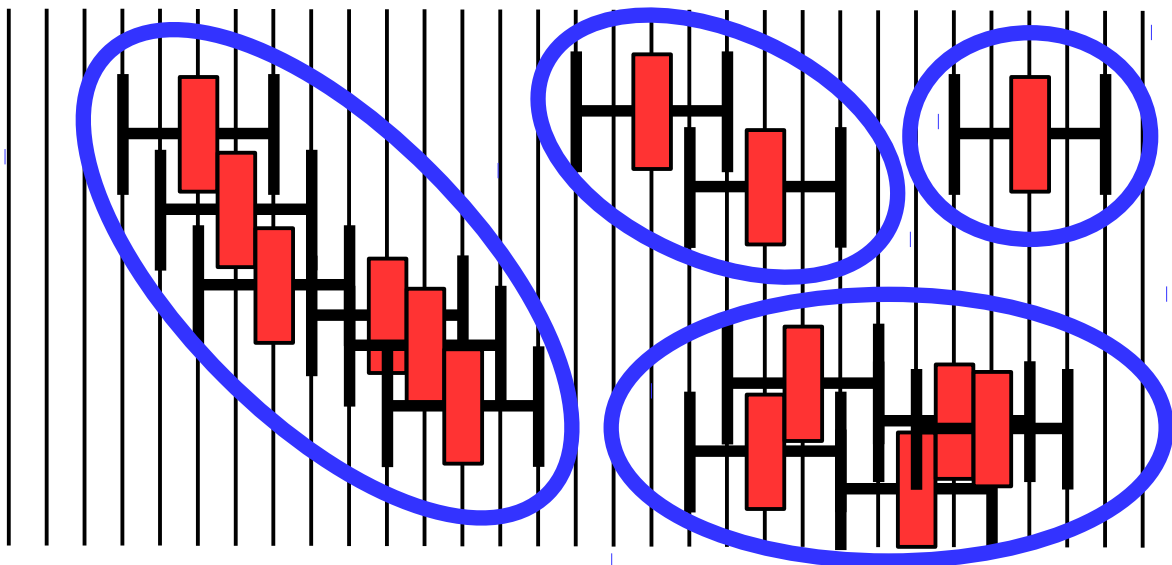
$$\begin{pmatrix} M_1(\omega) \\ M_2(\omega) \\ M_3(\omega) \\ M_4(\omega) \\ M_5(\omega) \\ M_6(\omega) \end{pmatrix} = \begin{pmatrix} R_0(\omega) & R_1(\omega) & 0 & 0 & 0 & 0 \\ R_1(\omega) & R_0(\omega) & R_1(\omega) & 0 & 0 & 0 \\ 0 & R_1(\omega) & R_0(\omega) & R_1(\omega) & 0 & 0 \\ 0 & 0 & R_1(\omega) & R_0(\omega) & R_1(\omega) & 0 \\ 0 & 0 & 0 & R_1(\omega) & R_0(\omega) & R_1(\omega) \\ 0 & 0 & 0 & 0 & R_1(\omega) & R_0(\omega) \end{pmatrix} \cdot \begin{pmatrix} S_1(\omega) \\ S_2(\omega) \\ S_3(\omega) \\ S_4(\omega) \\ S_5(\omega) \\ S_6(\omega) \end{pmatrix}$$

- ◆ Can invert this **Toeplitz** matrix in **$O(n \cdot \log(n))$** time (FFT)
 - Requires secondary FFT over wire number (n wires) as compared to nominal FFT over time (m time bins)
 - Total deconvolution time: **$O(m \cdot \log(m) \cdot n + n \cdot \log(n) \cdot m)$**
 - Still expensive calculation: reduce runtime with CalROI scheme

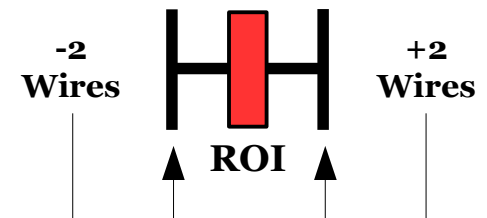


Merging of ROI's

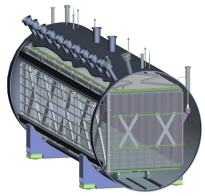
- ◆ Start with ROI's from nominal CalROI algorithm
- ◆ First step is to merge these into ROI clusters
 - **Expand each ROI** to span direct wire and N_{adj} adjacent wires
 - **Merge these units** into one cluster if they overlap
 - Scan entire wire plane over all time bins, merging left-to-right
 - If hit previously-merged ROI cluster, merge two ROI clusters into one
 - Relatively quick step since ROI's are sorted by wire number: **$O(n)$**



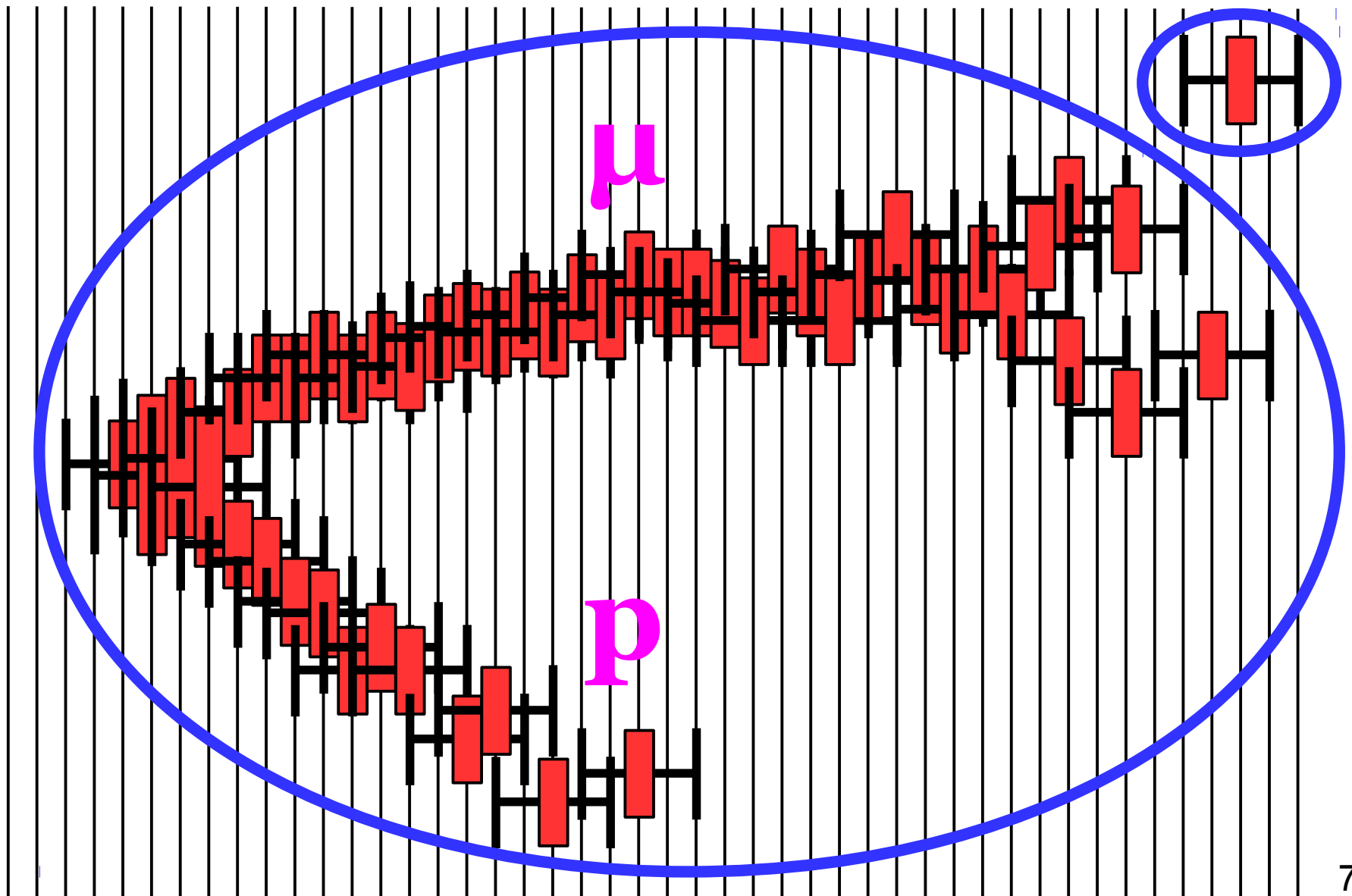
Example: $N_{\text{adj}} = 2$

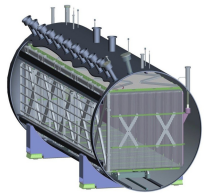


 **Merged ROI Cluster**

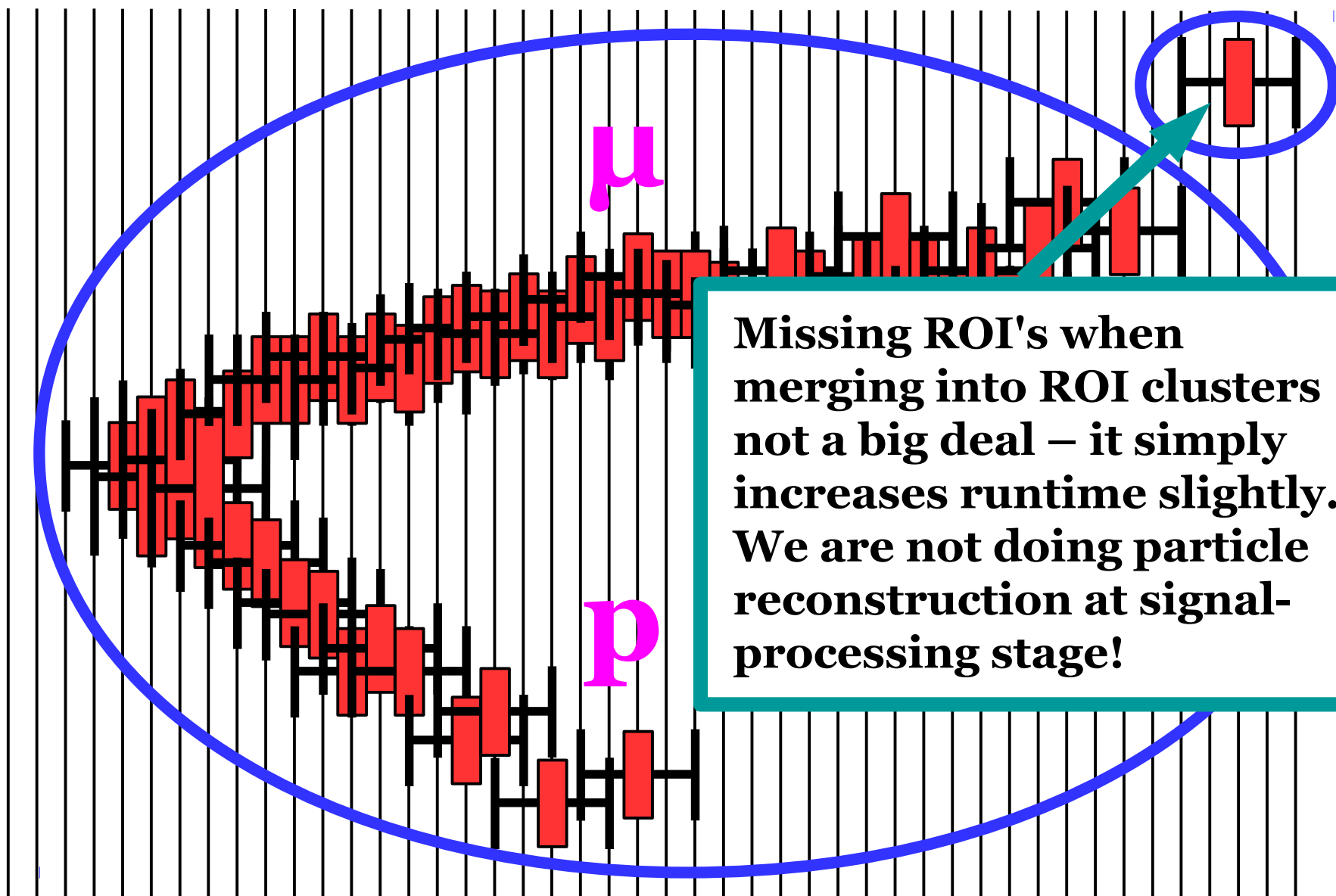


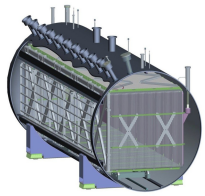
Merging of $\mu+p$ Topology





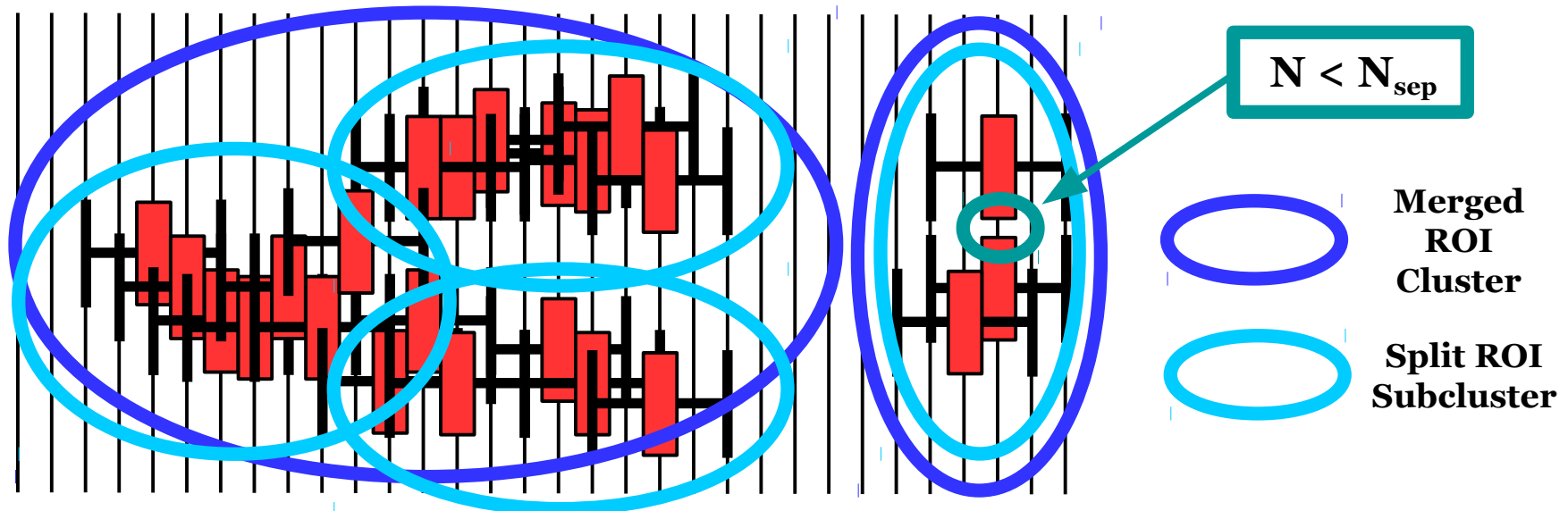
Merging of $\mu+p$ Topology

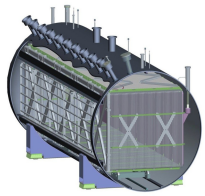




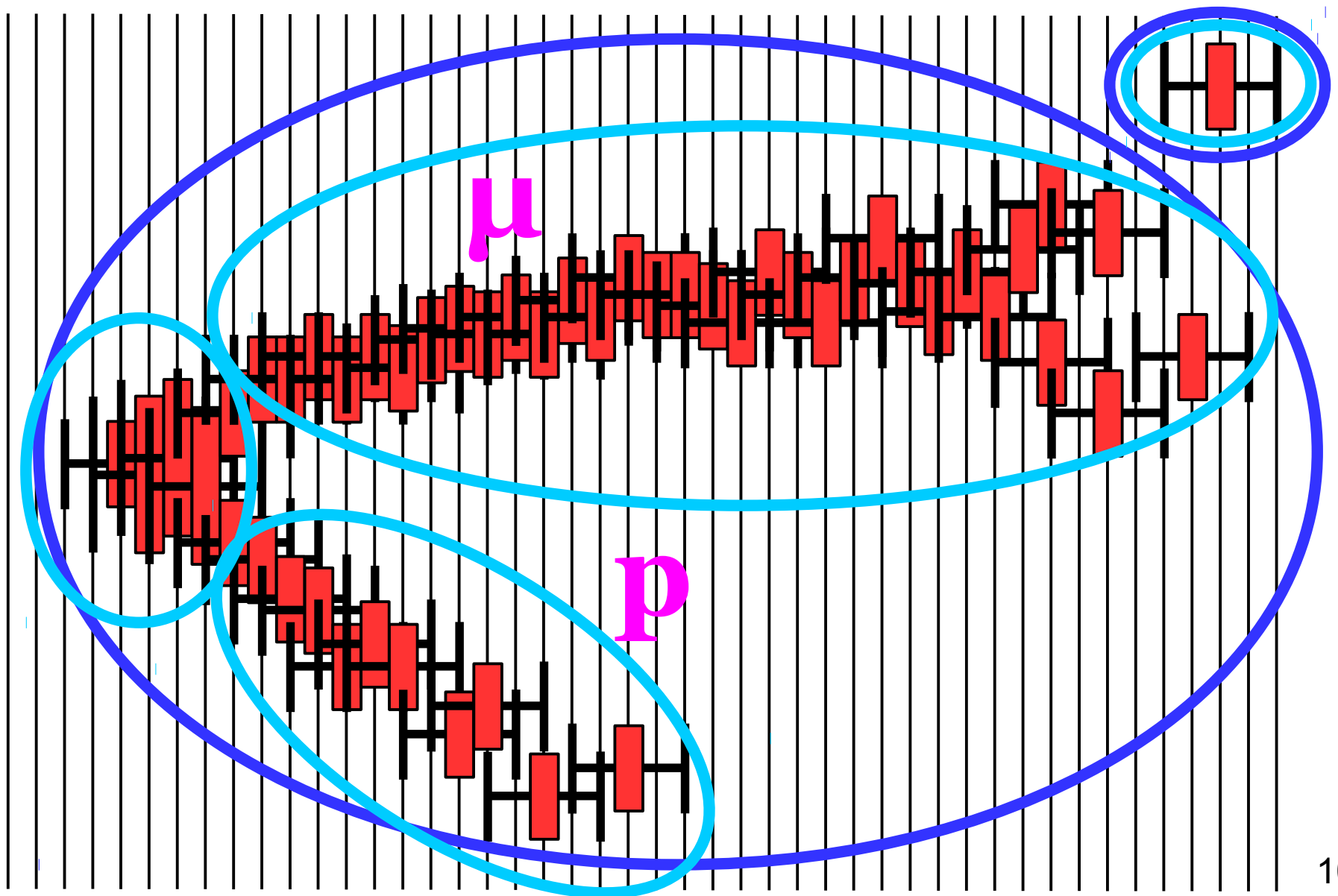
Splitting of ROI Clusters

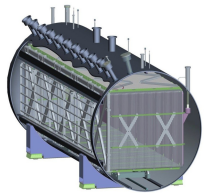
- ◆ Now we have a set of **ROI clusters** – could just make a huge rectangular FFT box (“ROI box”) around entire ROI cluster...
- ◆ ... but we can do better! Break into **ROI subclusters** (consistent with particle trajectories) to minimize runtime
 - Split based on number of ROI's within ROI cluster on a given wire
 - Introduces new parameter N_{sep} – minimal separation (in number of time bins) between two ROI subclusters of a given ROI cluster



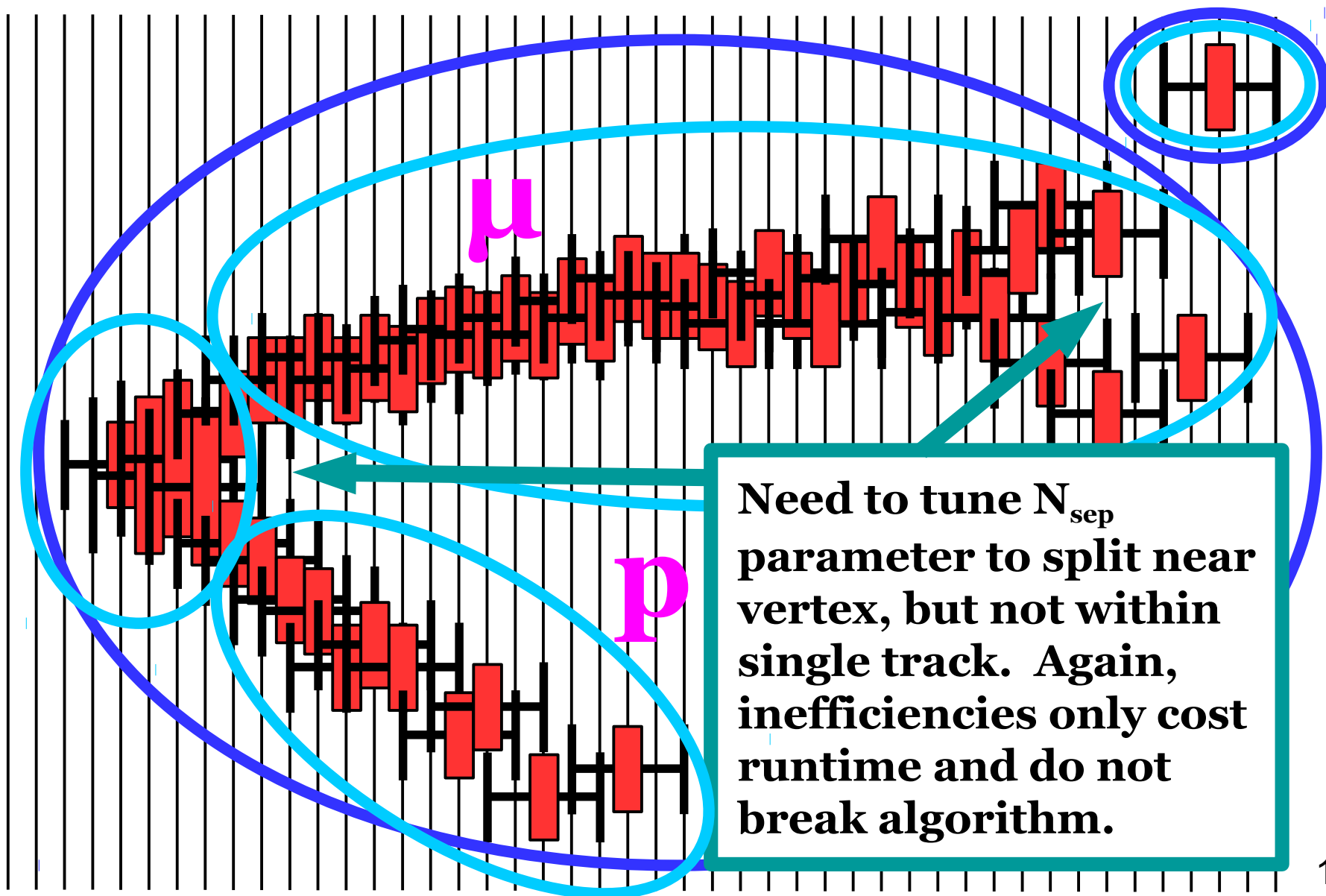


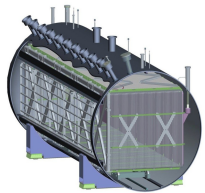
Splitting of $\mu+p$ Topology





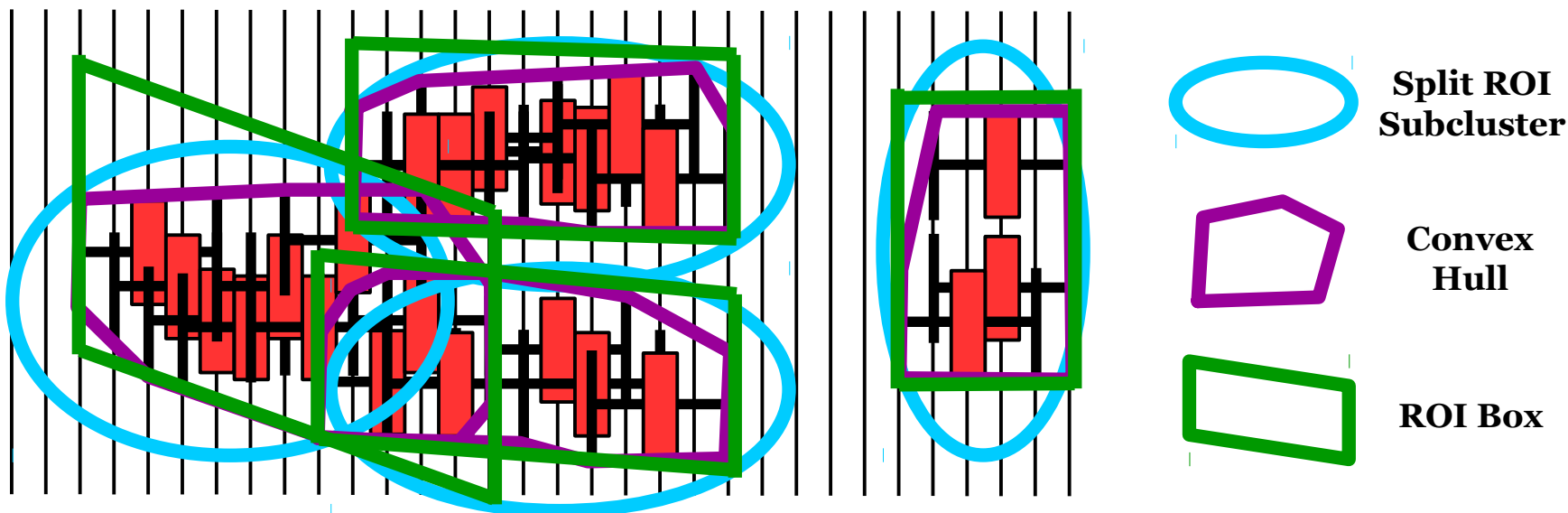
Splitting of $\mu+p$ Topology

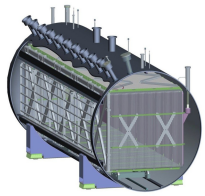




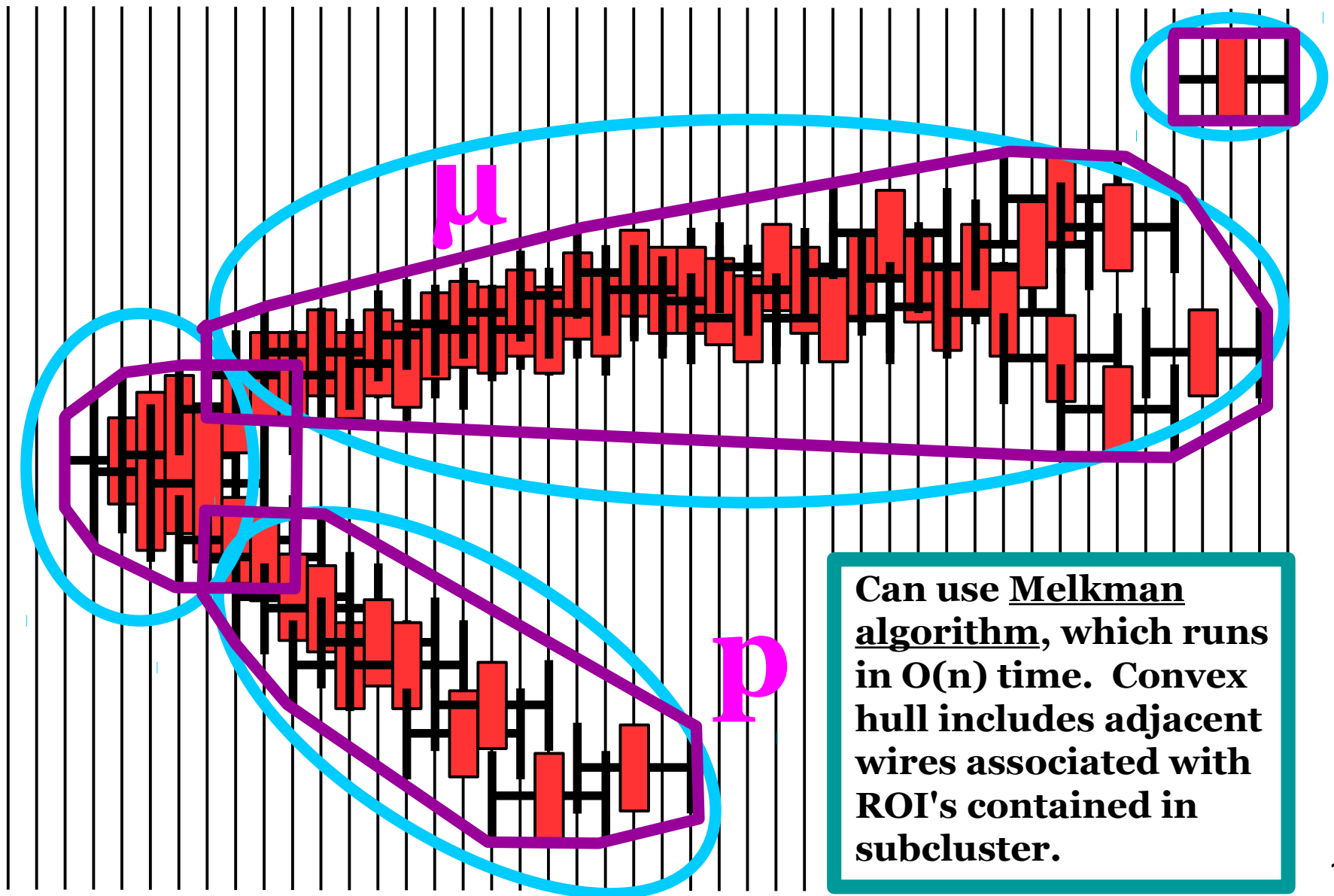
Forming of ROI Boxes

- ◆ Now we make minimal FFT window that surrounds all of the extended ROI units within our ROI subclusters – **ROI boxes**
 - ROI clusters and ROI subclusters are associations, while ROI boxes are parallelograms in wire-time space (geometric)
 - Same time window for every wire in ROI box, necessary for FFT
- ◆ Done in two steps for every ROI subcluster:
 - Form **convex hull** around extended ROI units within subcluster
 - Use convex hull to form **parallelogram of least area** (ROI box)

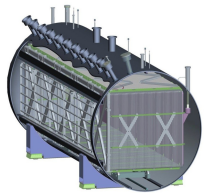




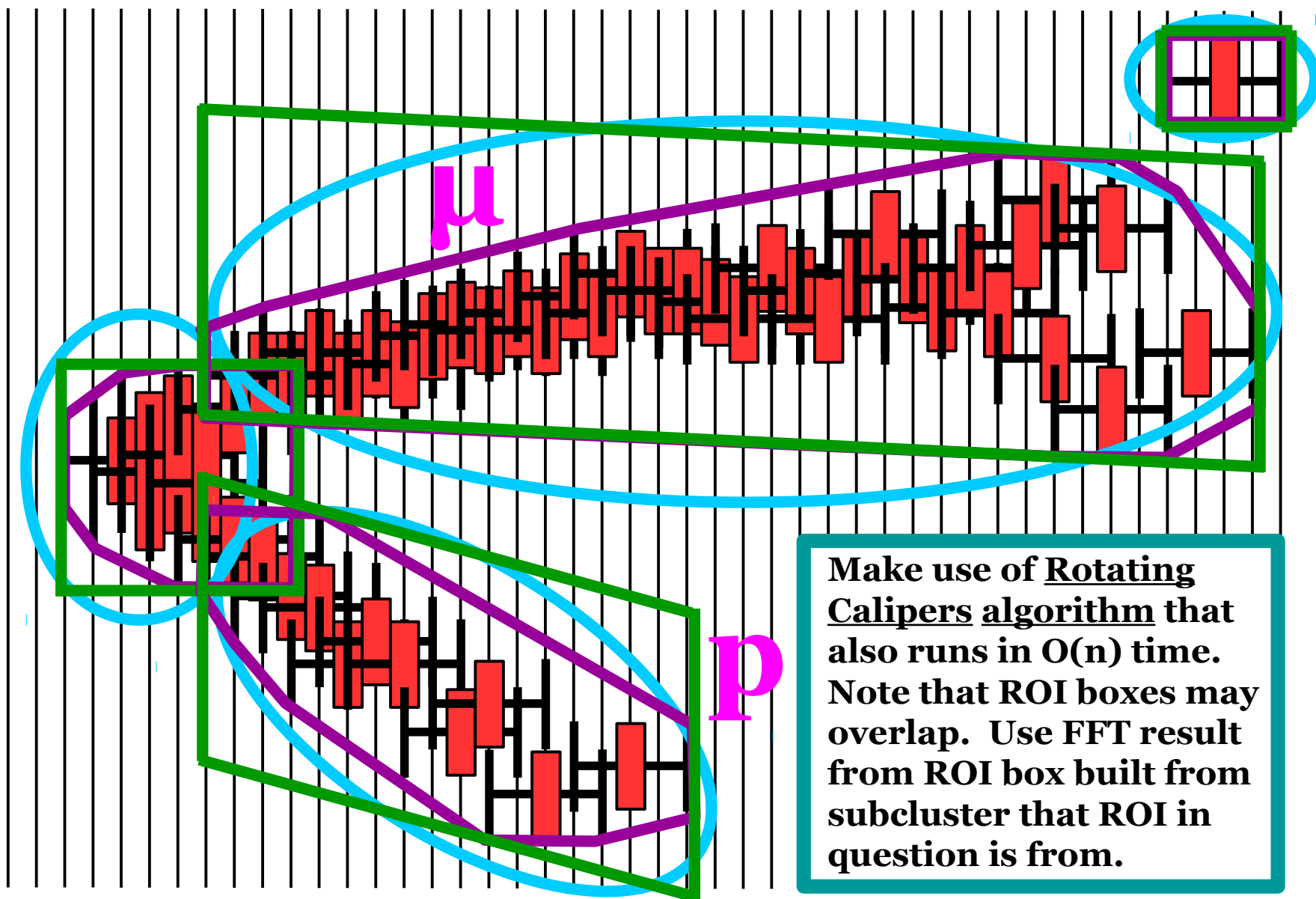
$\mu+p$ Topology: Convex Hulls



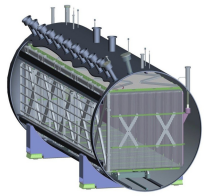
Can use Melkman algorithm, which runs in $O(n)$ time. Convex hull includes adjacent wires associated with ROI's contained in subcluster.



$\mu+p$ Topology: ROI Boxes

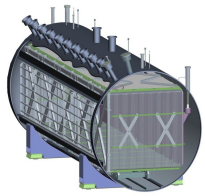


Make use of Rotating Calipers algorithm that also runs in $O(n)$ time. Note that ROI boxes may overlap. Use FFT result from ROI box built from subcluster that ROI in question is from.

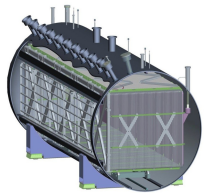


Summary

- ◆ Previously looked at deconvolution including induced charge response from adjacent wires for **CalWire** case
 - Expect CalWire deconvolution to be roughly **twice as slow**
- ◆ Now looking at first ideas for induced charge deconvolution suitable for **CalROI** – should be much faster
 - Factorized problem into three steps (to be done before FFT):
 - Merging of ROI's into “**ROI clusters**” – **$O(n)$** runtime (for ROI's already sorted by wire number)
 - Splitting of ROI clusters into smaller units consistent with track/shower candidates, “**ROI subclusters**” – **$O(n)$** runtime
 - For each ROI subcluster find appropriate FFT window across all associated wires, “**ROI boxes**” – **$O(n)$** runtime
- ◆ All steps seem reasonably quick, with straightforward implementation – main concerns are possible need for new filter, time-bin edge cases for ROI boxes, and tuning of N_{sep}



BACKUP SLIDES

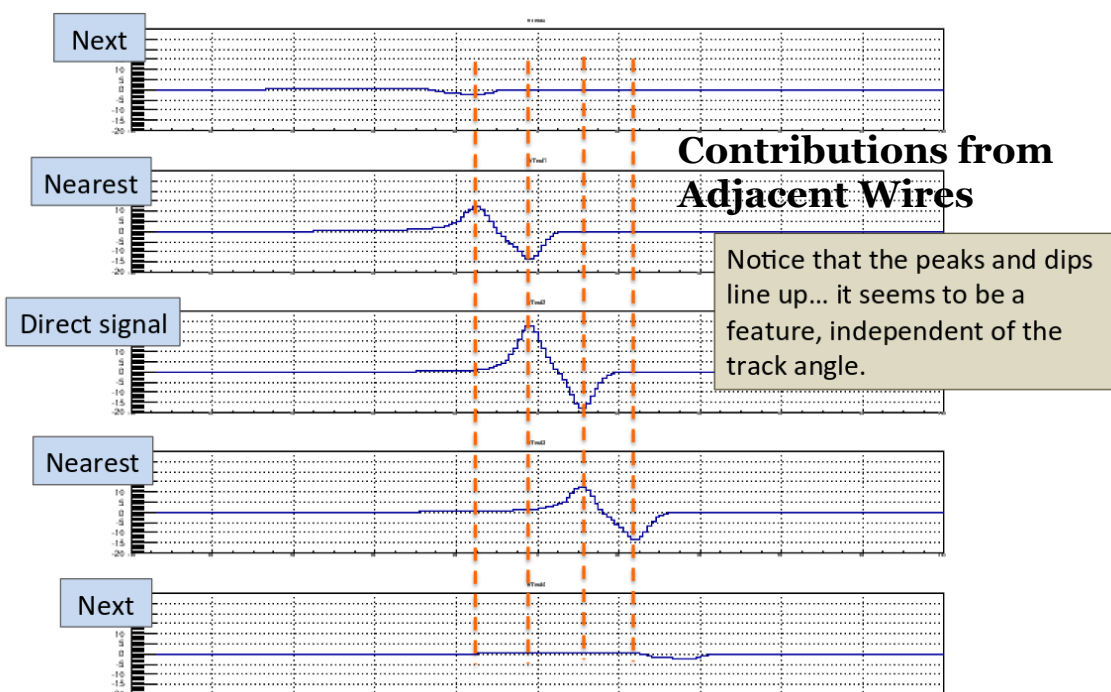
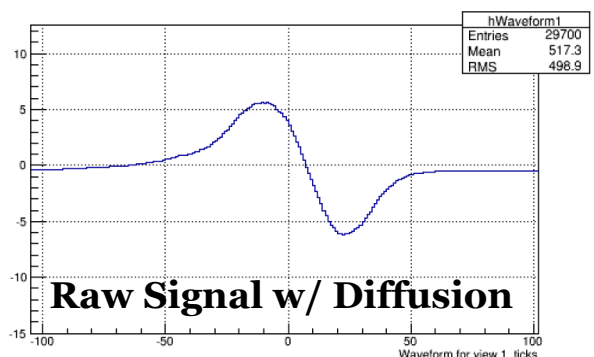
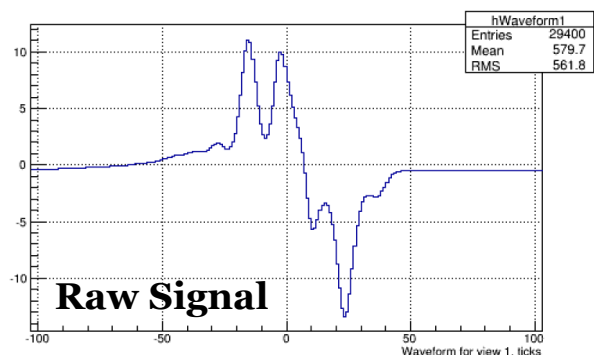


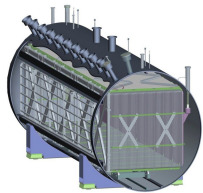
Motivation

- ◆ Has recently been pointed out (by Leon) that our field model neglects induced charge from adjacent wires
- ◆ **Important contribution** to raw signal waveform!
- ◆ Now accounted for in convolution – what about deconvolution?

L. Rochester

V Plane – 60° Track





Fast Fourier Transform

$$c_1 = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \cdot e^{-2\pi i \left(\frac{1}{T}\right)t} dt$$

...

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \cdot e^{-2\pi i \left(\frac{n}{T}\right)t} dt$$

1 signal of
16 points

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

2 signals of
8 points

0 2 4 6 8 10 12 14 1 3 5 7 9 11 13 15

4 signals of
4 points

0 4 8 12 2 6 10 14 1 5 9 13 3 7 11 15

8 signals of
2 points

0 8 4 12 2 10 6 14 1 9 5 13 3 11 7 15

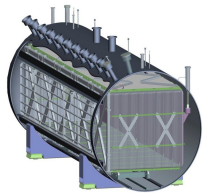
16 signals of
1 point

0 8 4 12 2 10 6 14 1 9 5 13 3 11 7 15

FIGURE 12-2

The FFT decomposition. An N point signal is decomposed into N signals each containing a single point. Each stage uses an *interlace decomposition*, separating the even and odd numbered samples.

$$\begin{pmatrix} M_{11} & M_{12} & \dots & M_{1n-1} & M_{1n} \\ M_{21} & M_{22} & \dots & M_{2n-1} & M_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ M_{n-11} & M_{n-12} & \dots & \dots & \dots \\ M_{n1} & M_{n2} & \dots & \dots & M_{nn} \end{pmatrix} \cdot \begin{pmatrix} f_1 \\ f_2 \\ \dots \\ f_{n-1} \\ f_n \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ \dots \\ R_{n-1} \\ R_n \end{pmatrix} \cdot (D_1 \ D_2 \ \dots \ D_{n-1} \ D_n) \cdot \begin{pmatrix} f_1 \\ f_2 \\ \dots \\ f_{n-1} \\ f_n \end{pmatrix}$$

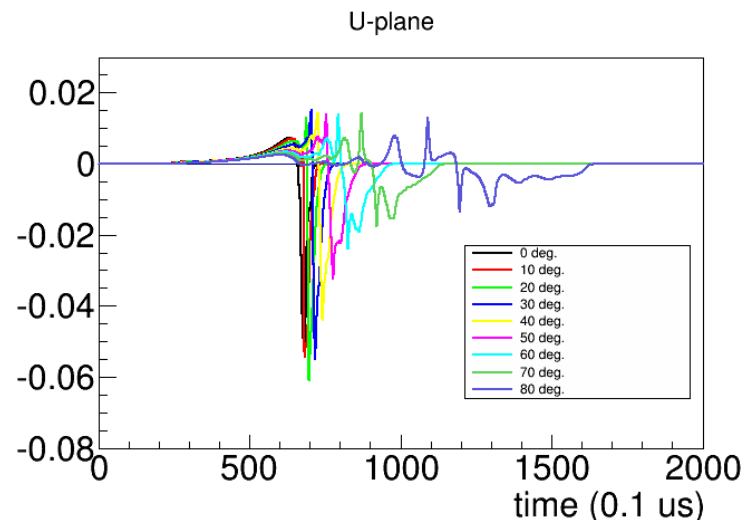
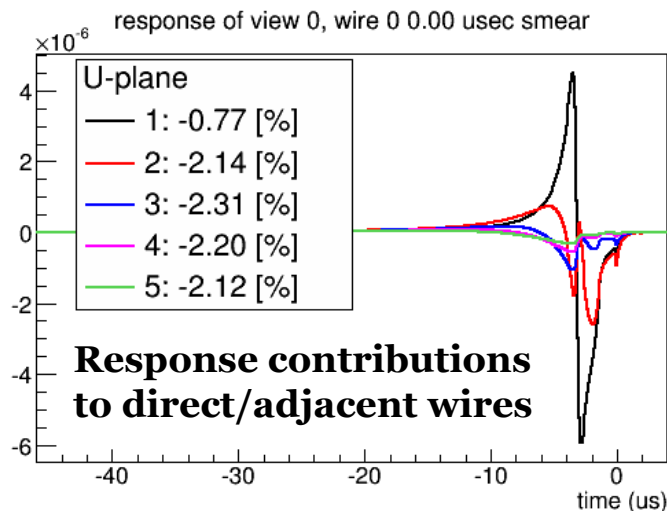


Induced Charge Response

- ◆ Use **toy MC** to study deconvolution – responses, raw signals below

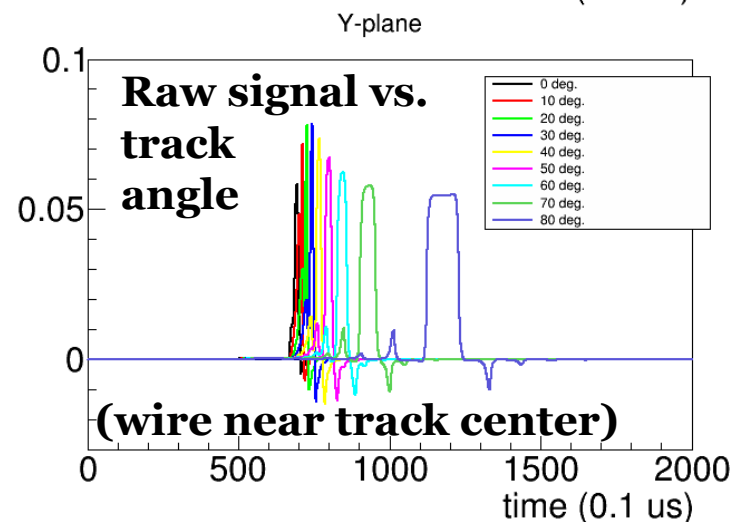
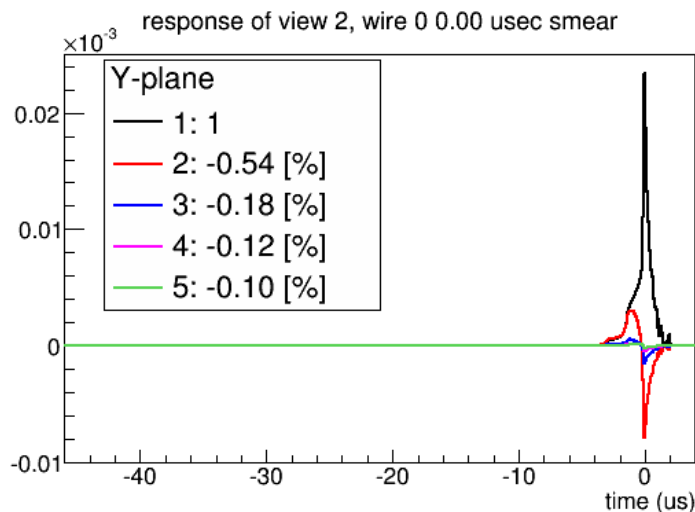
U

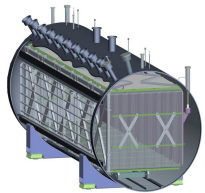
U



Y

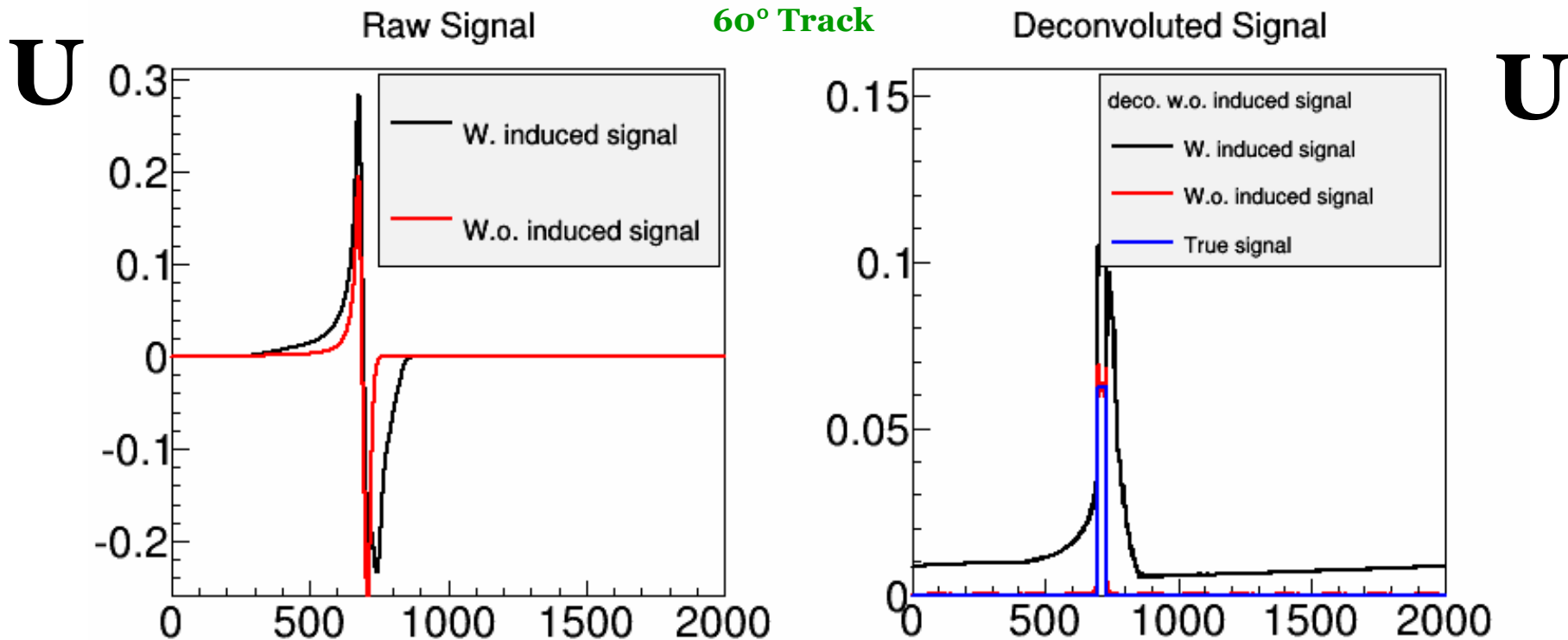
Y

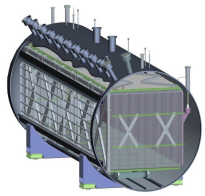




Impact on Deconvolution

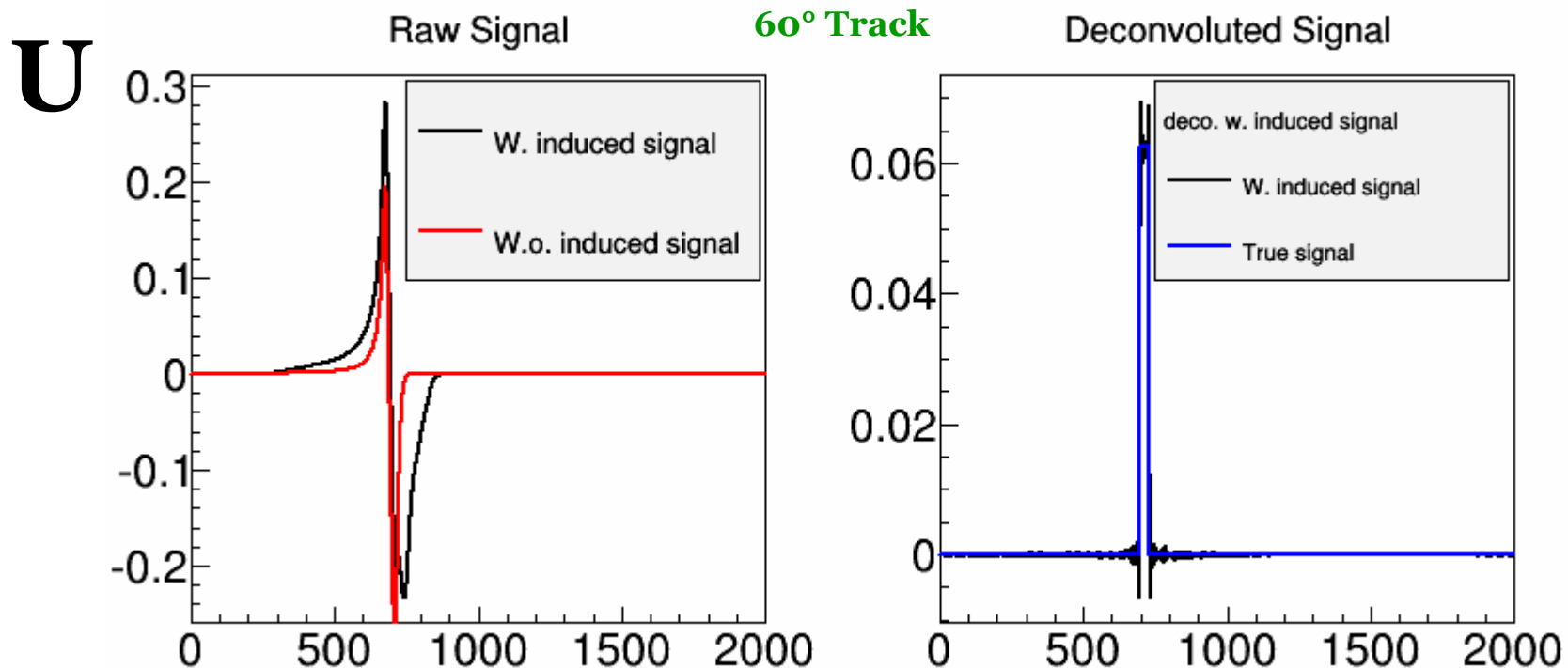
- ◆ Compare raw signal of **60° track** on **first wire** with and without induced charge signal from adjacent wires (left)
- ◆ **Neglect** indirect charge in deconvolution? Mismatch between convolution and deconvolution kernels leads to **problems** (right)

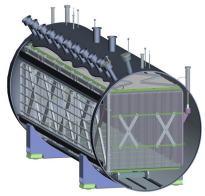




Results of New Scheme

- ◆ Make use of new indirect charge deconvolution scheme within toy MC (invert response matrix with secondary FFT over wire number)
 - Seems to **fix the problem** (see right)
 - Including very basic flat filter for now
 - Possibly need **another filter** for secondary FFT – investigating





CalWire vs. CalROI

- ◆ Studies with toy MC make use of CalWire-like setup: include all wires and all time bins in deconvolution
- ◆ New deconvolution scheme including induced charge roughly **twice as slow** compared to before
 - Before: $O(m \cdot \log(m) \cdot n)$
 - Now: $O(m \cdot \log(m) \cdot n + n \cdot \log(n) \cdot m)$
 - For m time bins, n wires
- ◆ What about CalROI-like setup?
 - Estimate seems to be closer to 50% slower on average
 - Additional overhead: first need to find **“ROI boxes”** to coordinate FFT over wire number – primitive tracking!

